IIP Wireless
Improving Internet Protocols for Wireless Links

Markku Kojo
University of Helsinki
Department of Computer Science

www.cs.helsinki.fi/research/iwtcp/

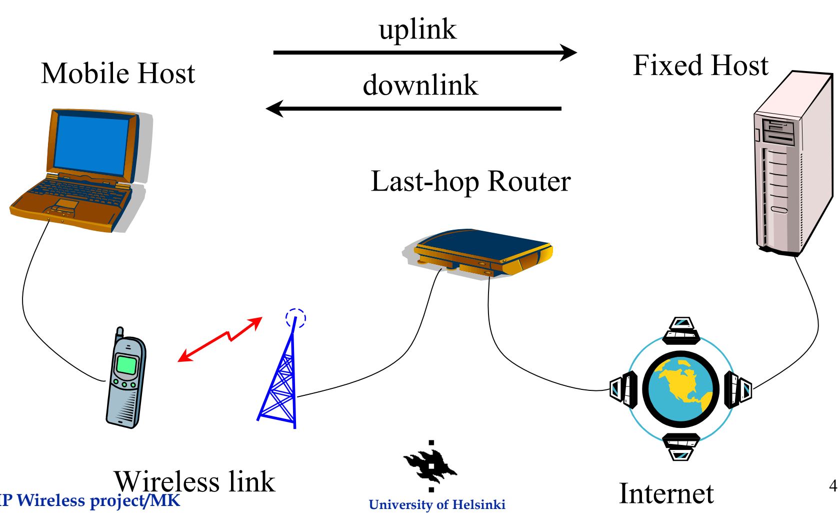
Presentation Outline

- Project Summary
- Environment of Interest
- Problem Areas
- Seawind Network Emulator
- TCP Enhancements
- Experiments and Example Results
- Conclusions

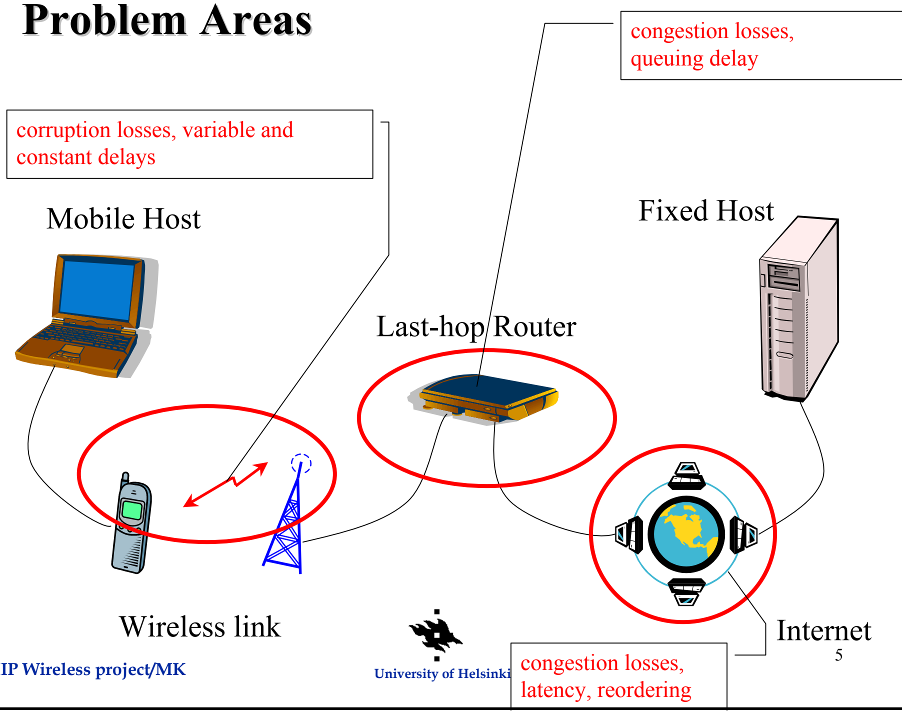
Project Summary

- Follow-up of IIP Mobile
- Partners: UHelsinki, Nokia Hungary, NRC, Sonera
- Topics
  - Empirical examination of TCP behaviour
  - TCP enhancements (implementations on Linux)
  - Development of a network emulator
  - Contributions to standardization
- Research continues in IIP Mixture?

Environment of Interest



Problem Areas



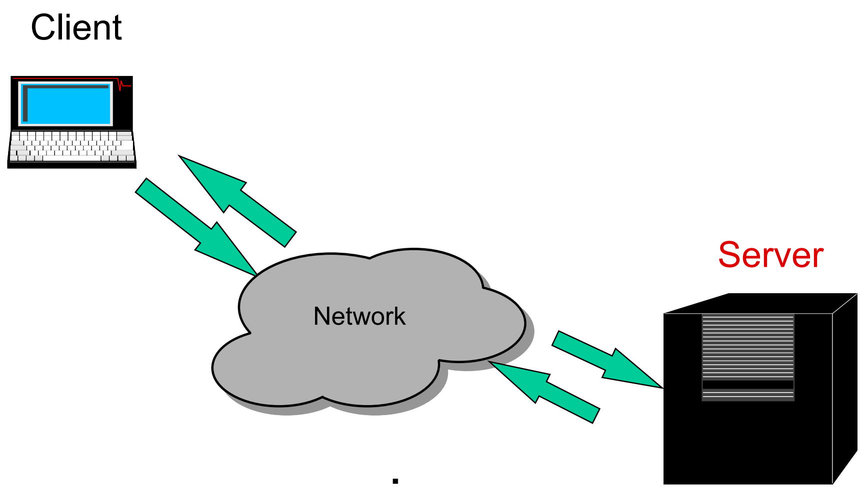
University of Helsinki

Software **E**mulator for **A**nalyzing **WI**reless **N**etwork **D**ata Transfers

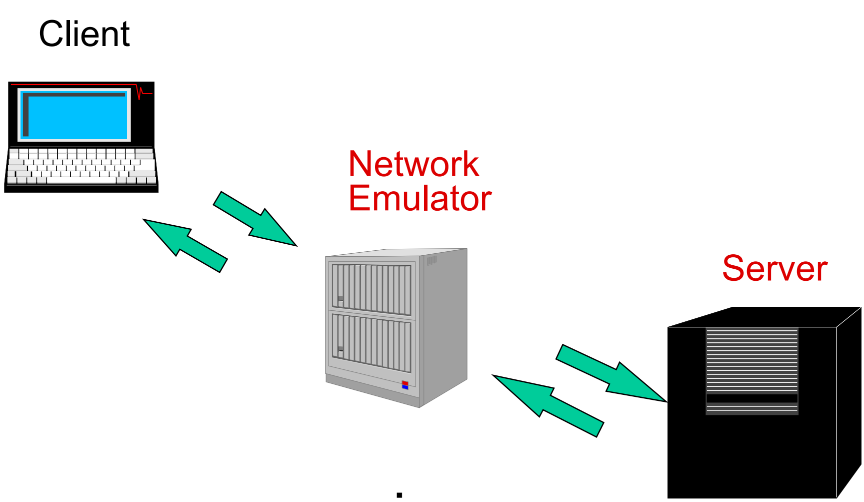
IIP Wireless Research Group
Department of Computer Science
University of Helsinki

University of Helsinki

Performance Measurements over a Real Network



Performance Measurements using Network Emulator



Objective of Seawind

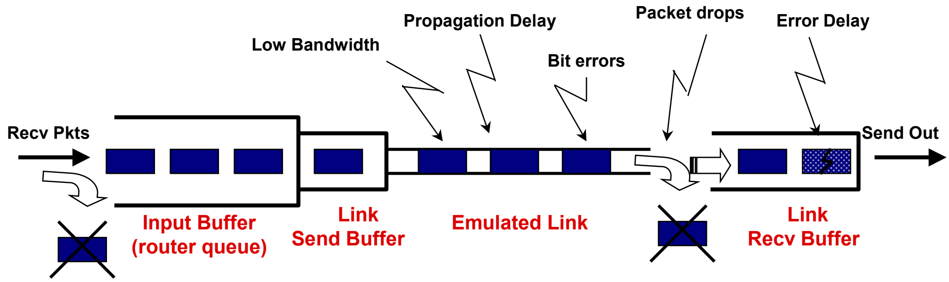
To implement a network emulator that allows performance measurements over an emulated network with specific characteristics using data traffic generated by a real-life application

- Enable studying and testing the behavior of higher layer protocols (transport/application) with real workload generated by a single user (e.g., a user as seen by a GPRS or GSM network)
- Enable studying possible interactions between the underlying network behavior and higher layer protocols
- Software implementation of a network emulator for Linux platform (extendable)
- Graphical User Interface and automated test run control
- Allow use of existing analysis tools, develop new analysis tools

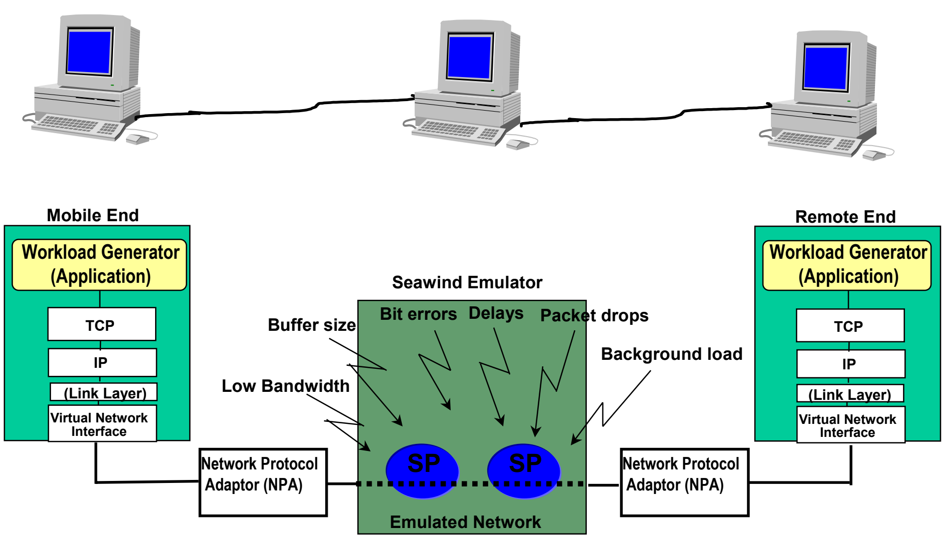
Wireless Network Characteristics of Interest

- Relatively low bandwidth
  (~4800bits/sec ➡ ~2 Mbits/sec)
- High latency (long round-trip time)
- Long, variable delays possible
- Data buffering along the "wireless path"
- Different error characteristics
  - error free, bit errors possible, data losses
- Link outages, intermittent connectivity
- Effects of background traffic

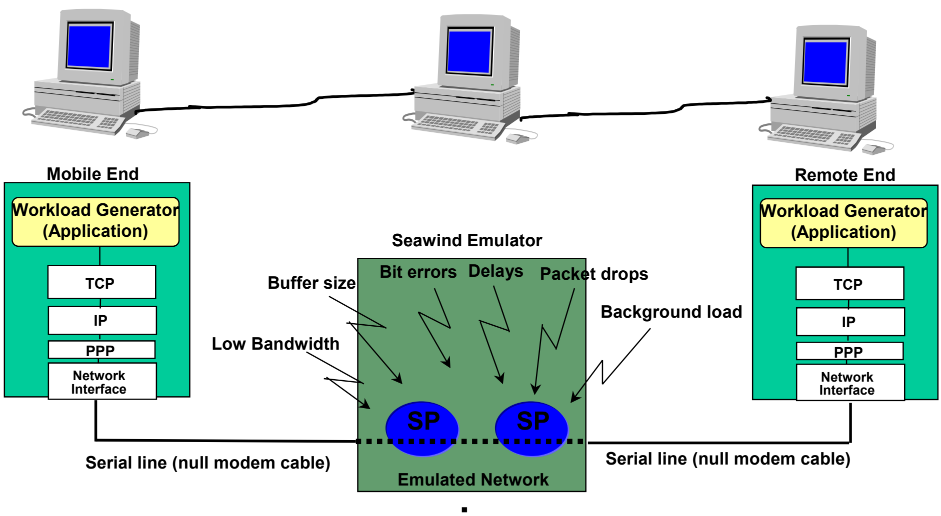
Network Emulation Model:
A Single Network Subsystem Emulated by a Seawind Simulation Process (SP)



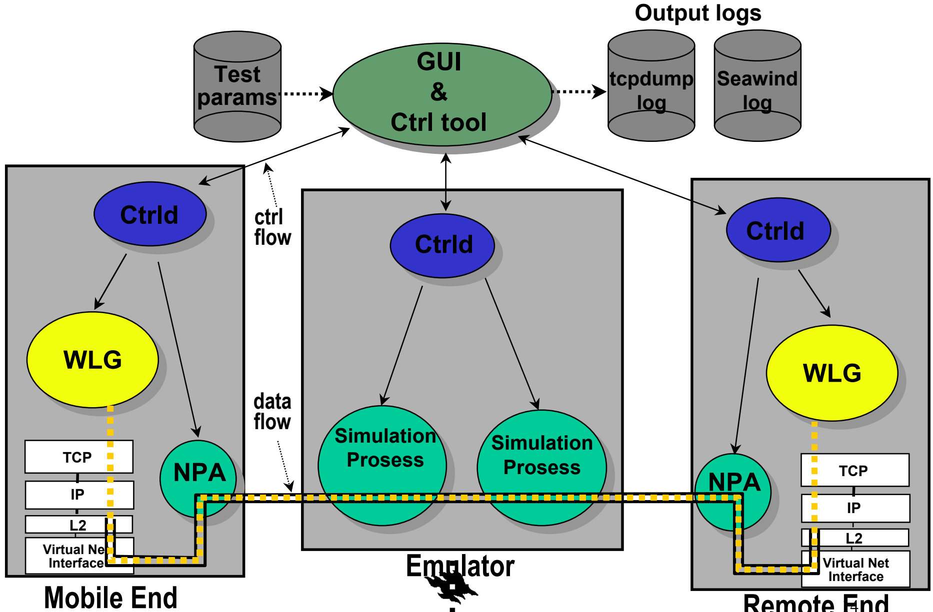
Structure of Seawind



Structure of Seawind (serial line version)



Software Components of Seawind

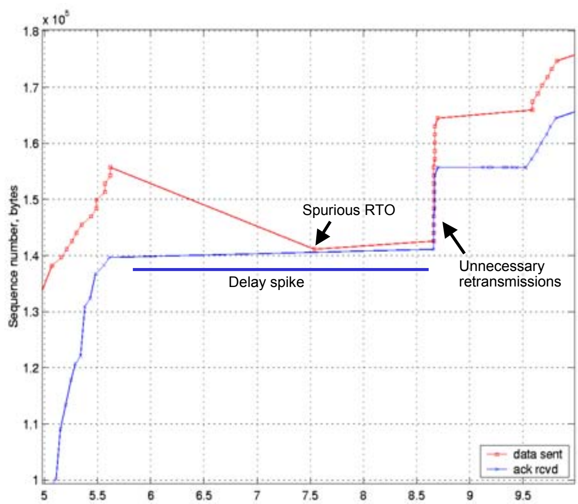


TCP Enhancements

- Baseline: NewReno, SACK
- Increasing the initial window
- Limited Transmit
- Random Early Detection (RED)
- Explicit Congestion Notification (ECN)
- TCP Eifel
- Forward RTO-Recovery

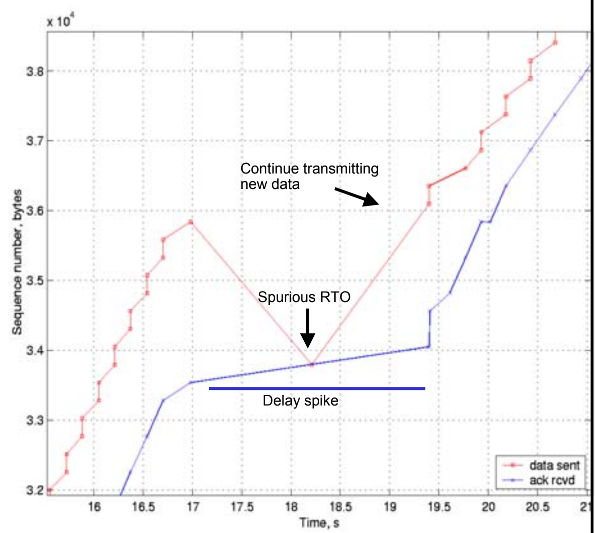
Spurious RTOs on Regular TCP

- Delay spikes occur on wireless networks
  - handoffs
  - link-layer error recovery
  - bandwidth variation
- Delay spike may trigger TCP retransmission timer
- Regular TCP sender retransmits all unacknowledged segments
- Eventually whole window is unnecessarily retransmitted in slow start
  - Network resources are wasted
  - Throughput gets worse



Delays and F-RTO

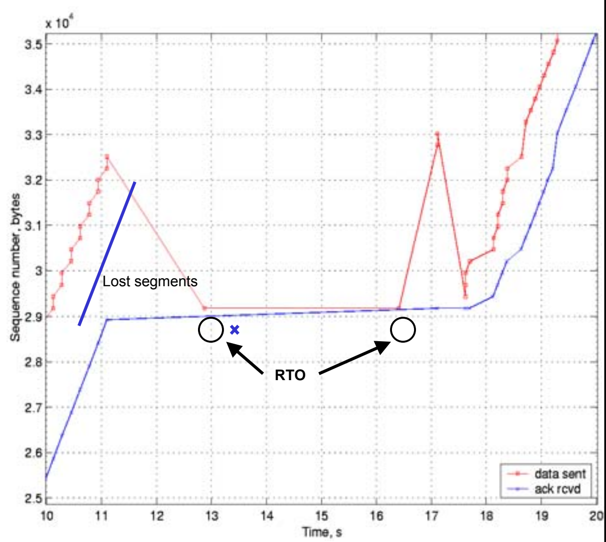
- When delay spike causes RTO to expire, the first unacknowledged segment is retransmitted
- 1st ACK acknowledges the retransmission: send 2 new segments
- 2nd ACK acknowledges data that was not retransmitted: RTO is declared spurious
- It is possible that second ACK does not advance window if
  - Packet following RTO was dropped
  - There was reordering following the RTO or a segment was duplicated



University of Helsinki

F-RTO and Data Loss

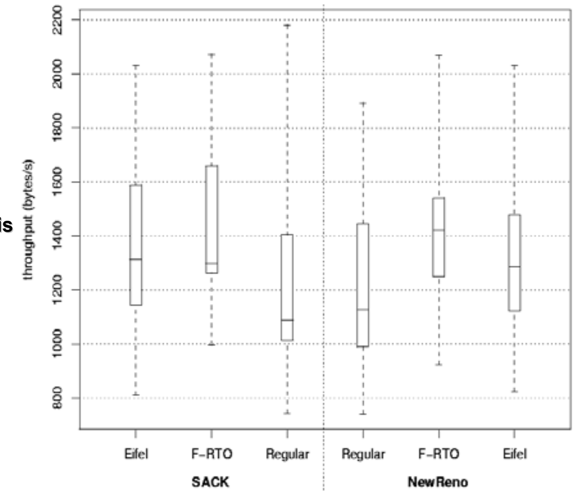
- When data is lost and RTO expires, F-RTO sender retransmits the first unacknowledged segment
- 1st ACK acknowledges the retransmission, so two new segments are transmitted
- 2nd ACK does not acknowledge new data, so the sender starts to retransmit unacknowledged segments in slow start
- When RTO is not spurious (i.e. it is caused by data loss), F-RTO is not worse than regular RTO recovery in any situation
  - Any combination of data and ACK losses should lead to slow start RTO recovery



University of Helsinki

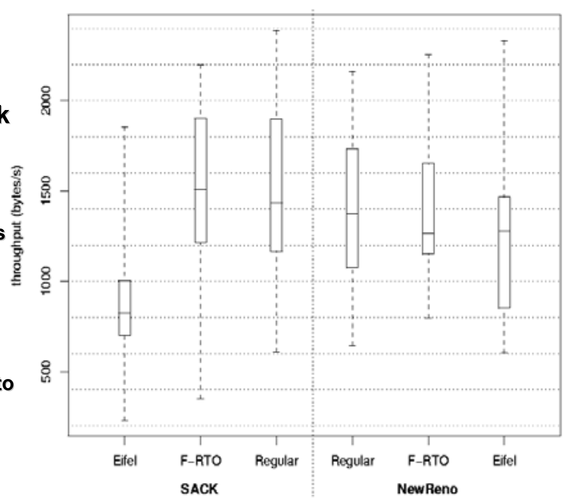
Performance on Delay Spikes

- Tests with Linux implementation on emulated slow wireless link
  - Link speed: 28.8 kbps
  - RTT: ~500 ms
  - MTU: 296 bytes
  - Delay spikes on random basis
- Both Eifel and F-RTO improve performance compared to regular TCP
  - Both avoid unnecessary retransmissions
- Eifel is not as good as F-RTO, because
  - TCP timestamps are additional overhead
  - There are more congestion-related losses



Performance on Data Loss

- Tests with Linux implementation on emulated slow wireless link
  - Link speed: 28.8 kbps
  - RTT: ~500 ms
  - MTU: 296 bytes
  - Link occasionally goes to loss state dropping all packets
- F-RTO performs about as well as regular TCP
- Eifel has problems
  - If first lost ACK corresponds to successfully transmitted data packet, sender continues by transmitting new data



Other Experiments

- Congestion on slow link with Traffic Mixes
  - Active Queue Management (RED)
  - Differentiated packet treatment
- Web-transfer performance
  - Typical Web-page sizes
- Traffic mixes on wireless link
  - TCP flows competing with UDP flows

Conclusions

- Wireless links are a challenge for TCP
- Significant improvements are possible
- Some of the proposed improvements work well but further enhancements are needed
- Empirical evaluation is necessary
- Seawind allows us to use real protocol implementations

More Information

HTTP://WWW.CS.HELSINKI.FI/RESEARCH/IWTCP/

EMail: Markku.Kojo@cs.Helsinki.FI
Kimmo.Raatikainen@cs.Helsinki.FI